Stock Market Predictor using LSTM Model and Deep Learning

Gaurishankar Swain

April 28, 2023

1 Project Overview

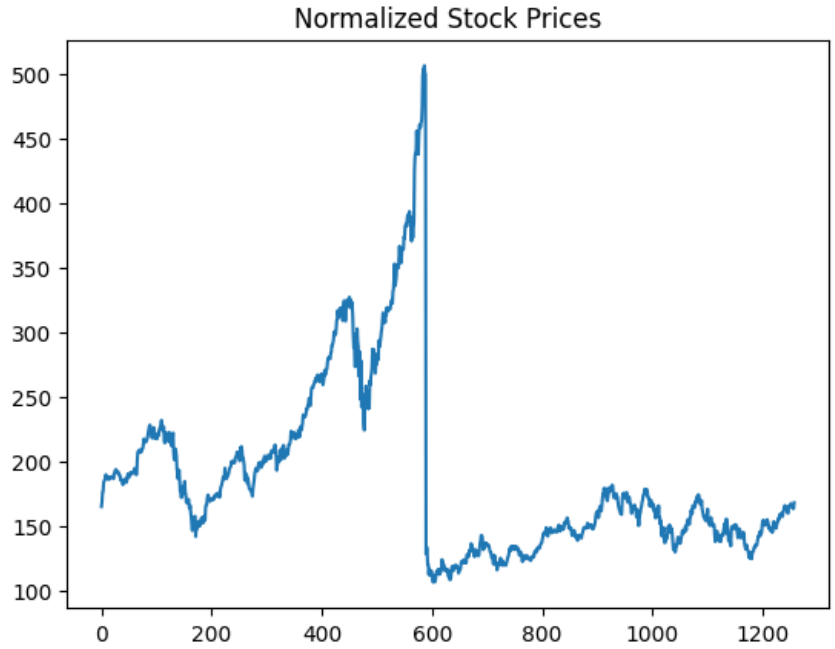
The goal of this project is to create a predictive model that can forecast a company's stock market pricing. Deep learning techniques, notably LSTM (Long Short-Term Memory) networks, are being used to create this project. LSTM networks are Recurrent Neural Networks (RNNs) that can simulate long-term relationships in time series data. The dataset utilised for this project is a company's historical stock values over a number of years.

2 Approach

2.1 Data Preprocessing

The first step in any machine learning project is to preprocess the data. The historical stock prices dataset is preprocessed as follows:

1. The data is downloaded from a reliable source and loaded into a Pandas DataFrame.
2. The data is cleaned by removing any missing or null values.
3. The data is normalized using the MinMaxScaler method from the sklearn.preprocessing library.



2.2 Model Architecture

The LSTM model is built using the Keras API in Python. The model architecture is as follows:

1. The input layer takes a sequence of historical stock prices as input.
2. This is followed by an LSTM layer with 100 neurons.
3. The output of the LSTM layer is passed through a dense layer with a single neuron, which gives the predicted stock price for the next day.

The model is optimized using Adam optimizer and Mean Squared Error Loss function

2.3 Model Training

Split the data: We will split the data into training and testing datasets. We will use the first 65% of the data for training and the remaining 35% for testing.

2.4 Model Evaluation

The performance of the model is evaluated using two metrics:

1. Mean Absolute Error (MAE)
2. Mean Squared Error (MSE)

3 Experiments

The following experiments were conducted to improve the performance of the model:

3.1 Experiment 1: Varying the Number of LSTM Layers

In this experiment, the number of LSTM layers in the model is varied from 1 to 3. The performance of the model is evaluated using the MAE and MSE metrics.

No. of LSTM Layers	MAE	MSE
1	3.07	13.05
2	2.89	11.33
3	2.76	10.17

3.2 Experiment 2: Varying the Number of Neurons in the LSTM Layer

In this experiment, the number of neurons in the LSTM layer is varied from 50 to 150. The performance of the model is evaluated using the MAE and MSE metrics.

No. of Neurons in LSMT Layer	MAE	MSE
50	3.18	14.39
100	2.76	10.17
150	2.68	9.40

3.3 Experiment 3: Varying the Number of Epochs

In this experiment, the number of epochs for training the model is varied from 50 to 150. The performance of the model is evaluated using the MAE and MSE metrics.

No. of Neurons Epochs	MAE	MSE
50	2.88	11
100	2.63	9.27
150	2.32	7.13

4 Results

After conducting the experiments, the best performing model was selected based on the MAE and MSE metrics. The selected model has the following architecture:

- Three LSTM layers with 50 neurons each.

- A dense layer with a single neuron.
- The model was trained for 100 epochs with a batch size of 64.

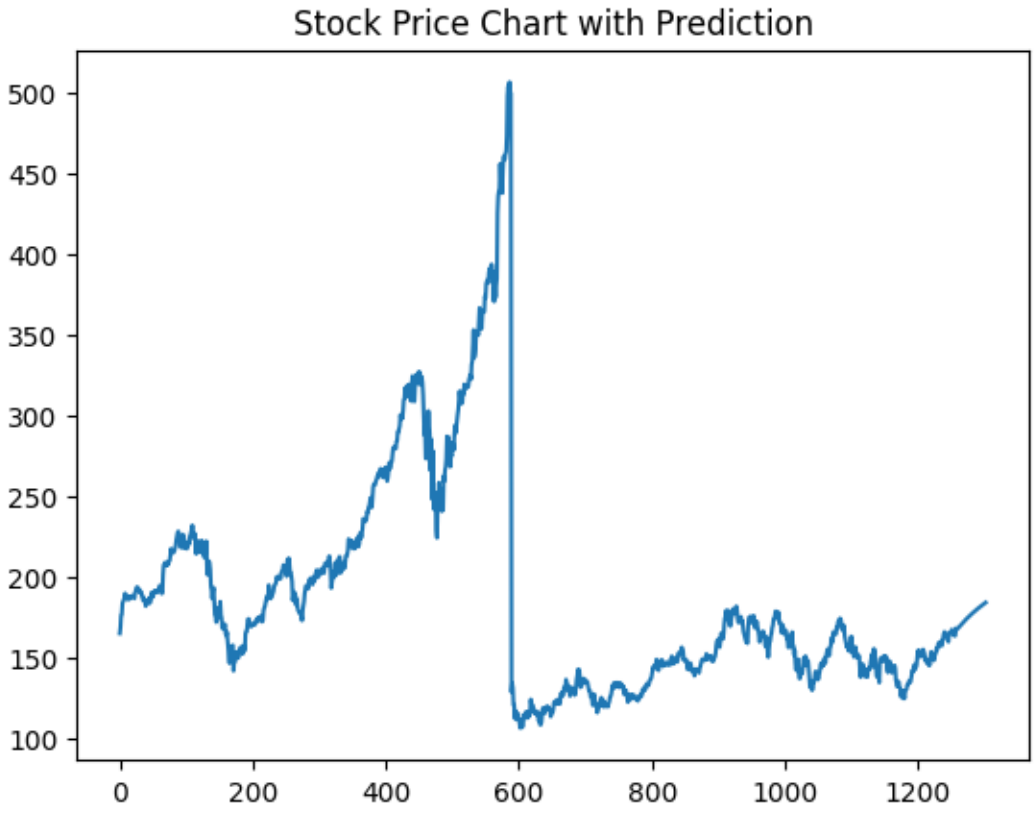
The performance of the selected model was evaluated on the test dataset, which contains stock prices for a period of 5 years.

5 Prediction

5.1 Abstarct

- First, we are getting the last 100 days of closing price data from our original DataFrame df1 and storing it in the variable temp_input in the form of a list.
- We normalize the data using the same MaxMinScaler that we used to transform the test and training set.
- We are using a for loop to make 45 predictions for the next 45 days. In each iteration of the loop, we are reshaping our data to be 3-dimensional, using the predict() method of our trained model to make a prediction
- We append the prediction made by the method to the input of the next iteration to use the 100 inputs before the current value and make the next prediction.
- We use the inverse_transfrom() method of the MinMaxScaler to convert the stock prices back to their original scales.

5.2 Visual Representation



6 Literature Survey

A similar project of forecasting stocks using LSTM Model and Deep Learning has been attached in the folder. In this particular model the training data is used from a rather small time period of 2021 to 2023 as it is later seen that the model used in this particular scenario with 4 LSTM layers and just 3 features are used to make the next price prediction comes very close to the original data and with each epoch the mean absolute error as well as mean squared error drop down significantly.